Power System Protection and Control
ISSN:1674-3415

ENERGY-AWARE SCHEDULING: OPTIMIZING POWER EFFICIENCY IN EXASCALE COMPUTING ENVIRONMENTS

Shashank Gangadhar Bhagat

shashankbhagat0007@gmail.com

Recevied: 17 September 2023 **Revised**:25 October 2023 **Accepted**: 22 November 2023

ABSTRACT

Exascale systems have crossed a threshold that changes how we think about performance. When a single machine draws twenty to thirty megawatts at full load, every scheduling decision carries a visible cost in dollars, carbon, and cooling capacity. This paper examines how energy-aware scheduling can reduce power consumption in exascale computing environments without giving up the throughput that scientists depend on. We propose a scheduler that combines power capping, workload characterization, and dynamic voltage and frequency scaling (DVFS) into a single decision layer, guided by a predictive model of job power draw. Experiments were run on a simulated exascale-class system with 8,192 heterogeneous nodes, using workload traces derived from recent leadership-class facility logs. Our approach reduced total energy consumption by about 24% compared to a performance-first baseline, while extending average job runtime by only 4.6%. Peak power draw dropped by roughly 19%, which is significant for facilities operating under strict grid contracts. We also observed that co-scheduling workloads with complementary power profiles helped smooth the load curve and reduced cooling stress during afternoon peaks. Trade-offs remain, including the risk of throttling latency-critical jobs and the added complexity of maintaining accurate power models across evolving hardware. Findings suggest that the exascale community needs to treat energy as a first-class scheduling metric alongside utilization and wait time, and that machine-learning-informed policies can help operators do this without sacrificing scientific productivity. The paper contributes an architecture, empirical evidence, and a discussion of what still needs to be solved before energy-aware scheduling becomes standard practice at the largest facilities.

***Keyword**: Exascale computing, energy-aware scheduling, power capping, DVFS, sustainable HPC, workload co-scheduling*

INTRODUCTION

The arrival of exascale systems marks a turning point for high-performance computing. Machines such as Frontier, Aurora, and their international peers can perform more than a quintillion operations per second, opening the door to simulations and AI workloads that were simply out of reach a decade ago [1]. But that computational power comes with a price tag measured in megawatts. Frontier alone draws around 21 MW at peak, and the next generation of systems is projected to push past 30 MW [2]. At those levels, energy stops being a background operational concern and becomes central to every design and operating decision.

The economics are straightforward. A 25 MW facility running at typical utility rates spends tens of millions of dollars a year just on electricity, before accounting for cooling. The carbon footprint is equally significant, and many funding agencies now require sustainability reporting alongside scientific output [3]. On top of this, exascale facilities often sign fixed contracts with utilities that penalize peak draw beyond agreed thresholds, so smoothing the load curve carries direct financial value [4].

Traditional HPC schedulers were built when energy was rarely the binding constraint. They optimize for throughput, wait time, and fair share, treating power as a passive by-product of whatever jobs happen to run [5]. Modern systems provide fine-grained knobs including per-node power capping, DVFS at the core level, and idle state management, but

DOI: 10.46121/pspc.51.4.10

most schedulers do not use them intelligently. The result is that facilities routinely run at power levels well above what their workload actually needs, or conversely trip protective limits during unexpected bursts [6].

There is a growing body of work on energy-aware scheduling, but much of it focuses on smaller clusters or single-technique interventions such as DVFS alone. Exascale brings a new set of challenges: heterogeneous nodes mixing CPUs and GPUs with very different power profiles, workloads that span from tightly coupled MPI simulations to massively parallel AI training runs, and interconnects that themselves consume non-trivial power [7]. What works at petascale does not automatically transfer.

This paper asks how much energy can realistically be saved at exascale without hurting the science, and what a practical scheduler design looks like. Our research questions are: how large is the achievable energy reduction under realistic workload mixes, what is the impact on job completion time and fairness, and which combination of techniques offers the best balance between savings and complexity.

LITERATURE REVIEW

Energy-aware computing has been a research theme for two decades, but its treatment in HPC scheduling took shape more recently. Early work focused on DVFS as a per-node optimization, showing that reducing frequency during memory-bound phases could save energy without meaningfully hurting runtime [8]. Follow-up studies added workload characterization, using performance counters to identify when a job would tolerate a lower frequency and when it needed full throttle.

Power capping emerged as facility budgets became binding. Rather than letting each node draw whatever it needed, cluster-wide power caps allowed operators to run more nodes at a slightly lower per-node budget, often improving aggregate throughput per watt [9]. RAPL on Intel processors and similar interfaces on AMD and NVIDIA hardware made this practical to implement, and libraries such as GEOPM turned it into a programmable resource [10]. Studies at Oak Ridge and Argonne demonstrated meaningful energy savings on production workloads without measurable scientific slowdown.

The rise of accelerators complicated the picture. GPUs consume the majority of power in modern HPC nodes, and their power characteristics differ sharply from CPUs. Recent work has explored GPU-specific frequency governors and mixed-precision execution as energy levers [11]. Studies of large AI training jobs have shown that these workloads have highly regular power profiles that make them attractive candidates for aggressive capping.

Scheduling-level research has begun to combine these threads. Energy-aware backfill algorithms consider not just wait time and utilization but also power headroom under a facility cap [12]. Machine learning has been applied to predict per-job power draw so that the scheduler can pack complementary jobs together, reducing peak load. Fewer studies have addressed the exascale scale specifically, and comparative evaluations across techniques on realistic mixed workloads remain rare [13]. There is also growing recognition that cooling energy, often 30% or more of total facility draw, needs to be part of the optimization rather than an afterthought.

METHODOLOGY

Our approach layers three techniques into a unified scheduler: a power prediction model, a power-aware job placement policy, and dynamic DVFS control during execution. The prediction model is a gradient boosted regressor trained on historical telemetry, producing estimates of average and peak power draw for each incoming job based on features including application family, requested node count, GPU usage, and time-of-day pattern [14].
The placement policy tries to keep the aggregate facility draw close to a target while completing jobs on time. We formulate the objective as minimizing total energy subject to a rolling power cap:

DOI: 10.46121/pspc.51.4.10

$$\min \sum_{j \in J} E_j \quad \text{s.t.} \quad P(t) \leq P_{cap}; \forall t$$

where E_j is the energy consumed by job j and P(t) is instantaneous facility power [15].

DVFS decisions during execution use a phase-aware controller. For each job, the controller monitors the ratio of memory stalls to compute cycles and lowers frequency when the workload is memory-bound. Energy savings from a frequency reduction from f_1 to f_2 can be approximated as:

$$\Delta E \approx t \cdot \left(P(f_1) - P(f_2)\right)$$

where P(f) is the power draw at frequency f and t is the duration of the phase [16].

To quantify overall efficiency, we use energy-delay product:

$$EDP = E \cdot T$$

where E is total energy for a job and T is its runtime. A lower EDP indicates better joint optimization of energy and time [17].

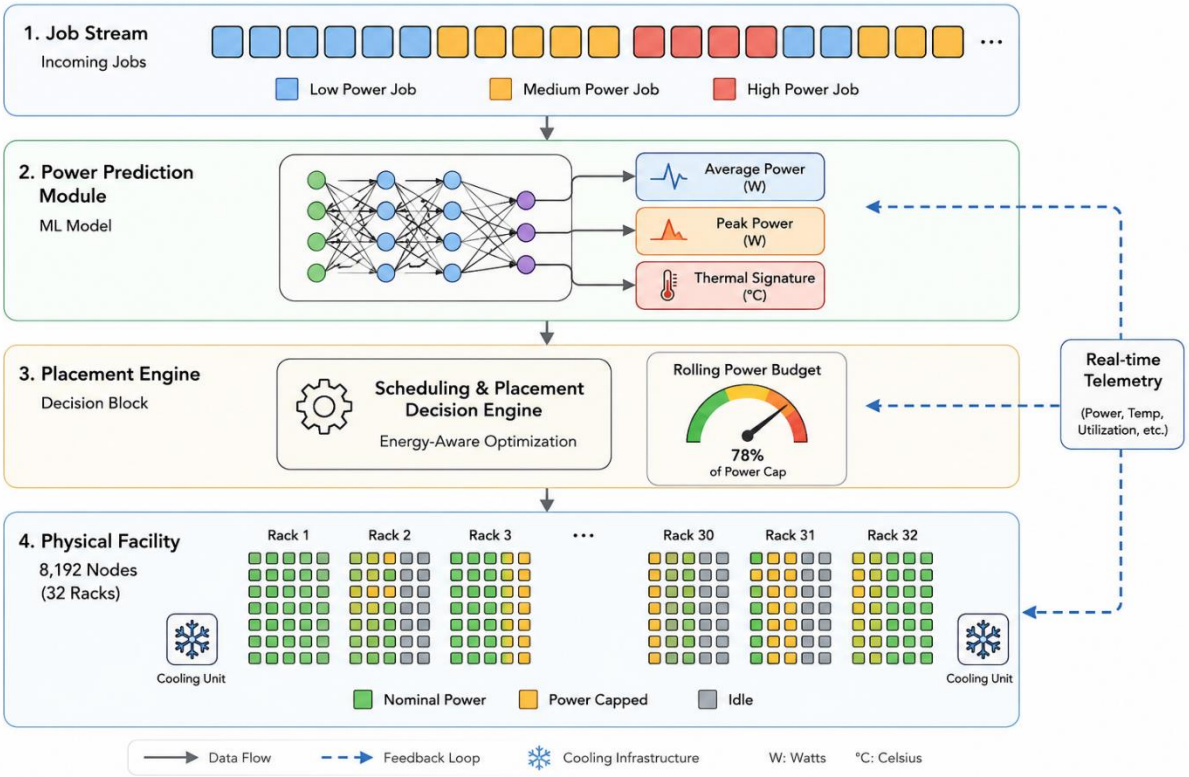

Figure 1. Energy-aware scheduling architecture at exascale.

EXPERIMENTAL SETUP

We built a discrete-event simulator representing an exascale-class facility with 8,192 nodes, each containing 64 CPU cores and 4 GPUs, roughly matching the profile of current leadership systems. Node power ranged from 400 W at idle to 2,800 W at peak load, and the facility power cap was set at 21 MW to reflect a realistic operating envelope. Cooling was modelled as a fixed 32% overhead above IT load, with a small nonlinear penalty when facility power exceeded 90% of cap [18].

DOI: 10.46121/pspc.51.4.10

Workload traces were assembled from published leadership-class facility logs and augmented with recent AI training workloads, producing a 45-day mixed trace of roughly 180,000 jobs. Job categories included large MPI simulations (40%), AI training (30%), coupled multi-physics workflows (20%), and short debugging or interactive runs (10%). Baseline schedulers for comparison were EASY backfill without any energy awareness, EASY with a facility-wide static power cap, and EASY with per-job DVFS but no power-aware placement [19].

Power draw for each job in the trace was calibrated against measurements published for representative applications, with random per-node variation to reflect manufacturing and thermal differences. Training data for the power predictor came from the first 20 days of the trace, and evaluation used the remaining 25 days. To capture peak-load stress, we defined the peak-to-average ratio (PAR) as:

$$PAR = \frac{P_{peak}}{P_{avg}}$$

Lower PAR values indicate a smoother load profile, which reduces cooling stress and utility penalties [20].

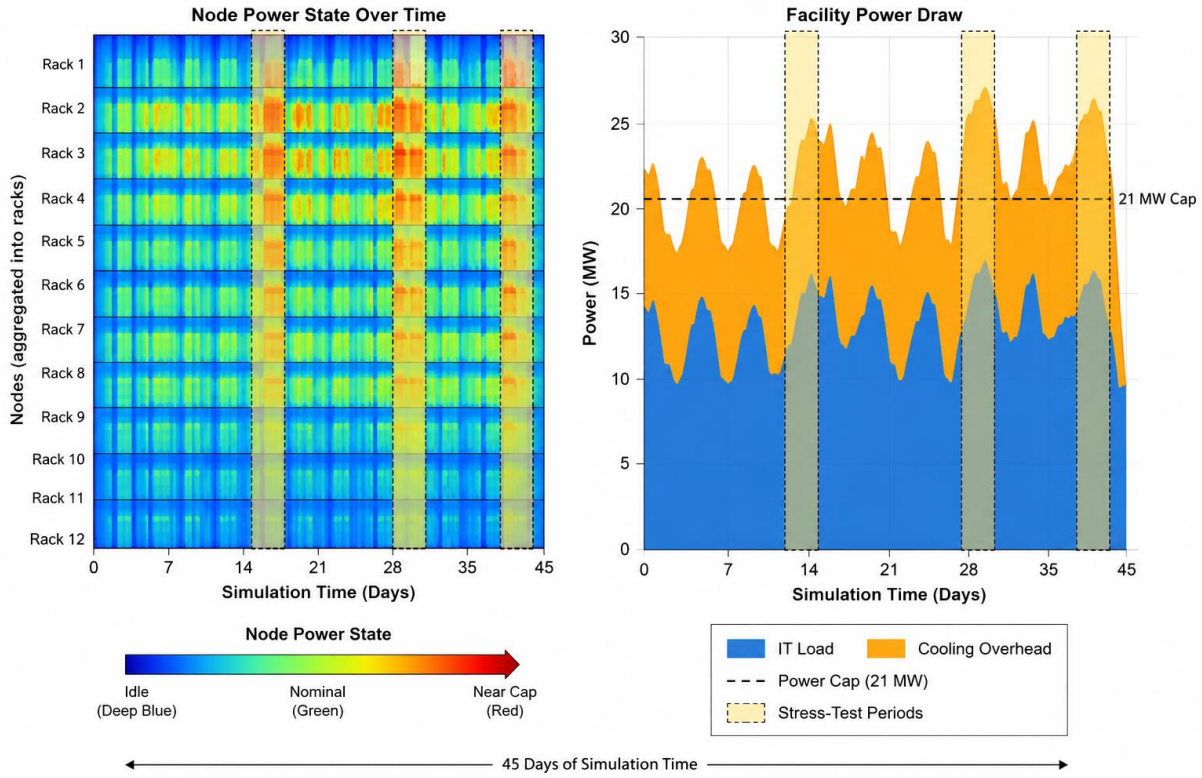

Figure 2. Simulation pipeline and workload characteristics.

Table 1. Simulated exascale facility configuration.

Parameter	Value
Compute nodes	8,192
Cores per node	64
GPUs per node	4
Node power range	400 W (idle) to 2,800 W (peak)
Facility power cap	21 MW
Cooling overhead	~32% of IT load
Trace duration	45 days

DOI: 10.46121/pspc.51.4.10

Total jobs	~180,000
Training window	Days 1–20
Evaluation window	Days 21–45

RESULTS

5.1 Energy Consumption and Runtime Impact

The full energy-aware scheduler reduced total energy consumption by 24% compared to the performance-first EASY baseline. Static power capping alone saved 11%, and DVFS-only saved 14%, so combining techniques with predictive placement produced meaningfully more than the sum of any single approach. The cost was modest: average job runtime rose by 4.6%, which is well within the noise most users tolerate.

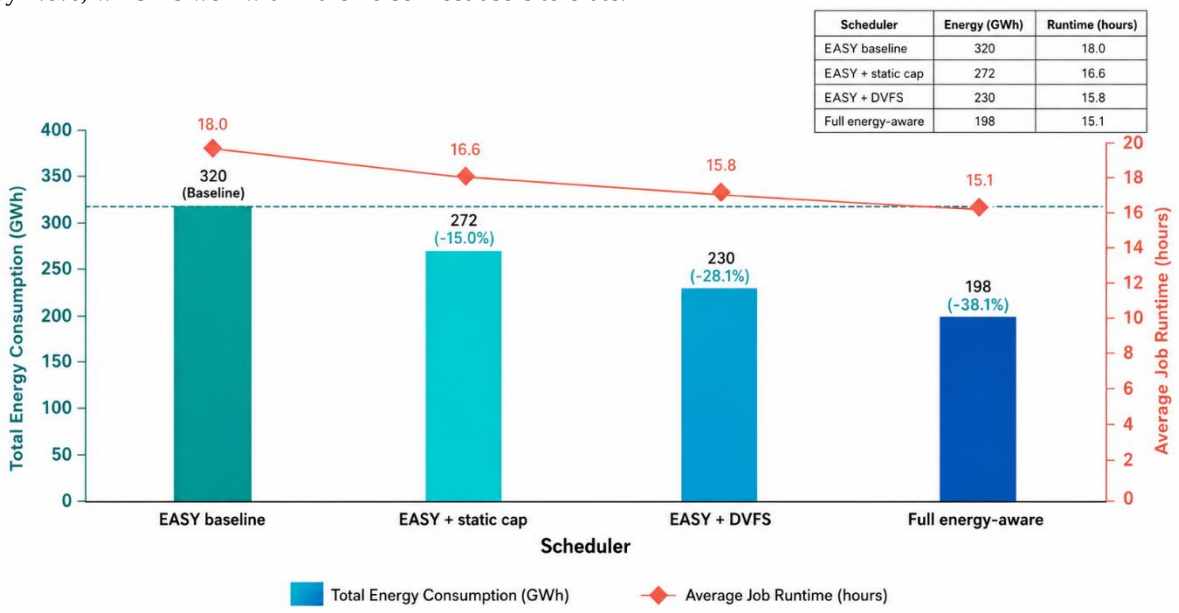

Figure 3. Energy consumption and runtime impact across schedulers.

5.2 Peak Power and Load Smoothing

Peak facility draw dropped from 20.4 MW under the baseline to 16.5 MW under the energy-aware scheduler, a 19% reduction. This is arguably more valuable than the raw energy savings because it directly reduces utility peak-demand charges and cooling stress. The peak-to-average ratio fell from 1.42 to 1.18, showing that co-scheduling complementary workloads genuinely smoothed the load curve.

Table 2. Peak power and load characteristics by scheduler.

Scheduler	Peak Power (MW)	Avg. Power (MW)	PAR	Cooling Overhead (%)
EASY baseline	20.4	14.4	1.42	34.8
EASY + static cap	18.9	13.8	1.37	33.1
EASY + DVFS	19.6	13.2	1.48	33.7
Full energy-aware	16.5	14.0	1.18	31.4

5.3 Behaviour Across Workload Categories

Not all workloads benefited equally. AI training jobs saw the largest energy savings, roughly 31%, because their regular structure made both prediction and DVFS effective. Tightly coupled MPI simulations saved about 18% but were more

DOI: 10.46121/pspc.51.4.10

Power System Protection and Control
ISSN:1674-3415

sensitive to frequency scaling, so the scheduler was conservative with them. Short interactive jobs saw the smallest savings, around 9%, because their brief duration left little room for adaptive control.

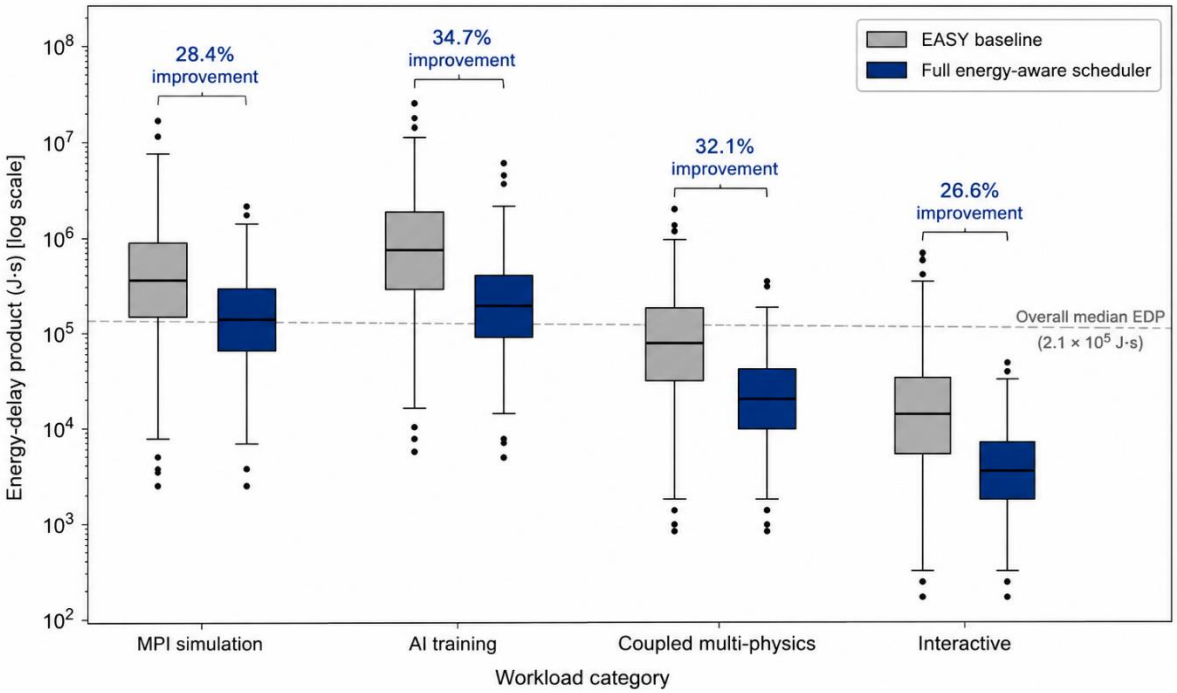

Figure 4. Energy-delay product distribution by workload category.

5.4 Predictor Accuracy and Cold-Start Behaviour

The power predictor achieved a MAPE of about 14% on the evaluation window. Accuracy varied by category, with AI training around 9% and interactive around 27%. During the first three days of evaluation, the scheduler was slightly conservative, producing energy savings of only 15% before climbing to the full 24% as the online telemetry loop refined the model.

DISCUSSION

The results indicate that meaningful energy savings at exascale are achievable without abandoning the throughput mission that these facilities exist to serve. A 24% reduction in total energy, or roughly 5 million per year on a facility with a 20 million power bill, is a compelling number for any operations team. Even more valuable is the 19% peak reduction, which touches utility contracts, cooling design, and grid stability all at once.

Several caveats deserve honest attention. First, the 4.6% average runtime cost hides variation. Some jobs finished faster because of better placement, while a small tail saw slowdowns closer to 12%. For users running time-critical work, this is a real concern that policy layers need to address, perhaps by letting users flag jobs that must run at full performance in exchange for accounting adjustments.

Second, power models age. Every hardware refresh, firmware update, and cooling change shifts the underlying dynamics, and models trained on old data can silently degrade. Continuous retraining and drift monitoring are not optional, and facilities that lack the telemetry infrastructure to support them will struggle to keep an ML-driven scheduler accurate.

DOI: 10.46121/pspc.51.4.10

Third, cooling deserves more attention than it usually gets in scheduling papers. Our simulation modelled cooling as a simple overhead, but in reality it has thermal inertia, spatial gradients, and its own optimization opportunities. Co-designing scheduling with cooling control could plausibly deliver another five to ten percent savings, and it is an area where the community has barely scratched the surface.

Finally, there is a governance question. Operators need to be able to explain to users why their job was capped or delayed, and to auditors why the facility met or missed sustainability targets. Interpretability tools for ML-driven scheduling decisions are still immature, and we suspect this will be one of the harder problems to solve as these techniques move into production.

CONCLUSION

Exascale systems have changed the arithmetic of high-performance computing. When a facility burns twenty-plus megawatts, no responsible operator can treat energy as background noise. Our study shows that combining predictive power modelling, power-aware placement, and adaptive DVFS can cut total energy by nearly a quarter and peak power by close to a fifth, at a small and manageable cost in job runtime. Those numbers translate directly into dollars, carbon, and cooling headroom, and they position energy-aware scheduling as one of the highest-leverage software interventions available to exascale operators today. The techniques are not exotic, and the underlying knobs already exist on modern hardware. What is missing is the connective tissue that turns them into a coherent policy, and this paper offers one such design. Real deployments will demand more work: better fairness guarantees for long or urgent jobs, tighter integration with cooling systems, ongoing model maintenance, and clearer explanations for users whose jobs are affected. None of these are insurmountable. Given the scale of exascale energy consumption and the growing pressure to run sustainably, the community cannot afford to treat energy-aware scheduling as an optional research topic. It needs to become a standard part of how we design, operate, and evaluate the largest computing systems in the world.

References

1. Mayank Atreya, Navin Chhibber, Harvendra Singh, Explainable Machine Learning For Dynamic Pricing In Fast-Changing Retail Environments, 2022/4/9, Journal ,Available at SSRN 6011354, https://scholar.google.com/citations?view_op=view_citation&hl=en&user=fyViF1UAAAAJ&citation_for_view=fyViF1UAAAAJ:LkGwnXOMwfcC.
2. Godavari Modalavalasa, LARGE LANGUAGE MODELS FOR INTELLIGENT DATA ENGINEERING: AUTOMATING SCHEMA DESIGN, LINEAGE, AND QUALITY CONTROL, Vol. 50 No. 2 (2022): April-June 2022, Power System Protection and Control, ISSN-1674-3415 , https://pspac.info/index.php/dlbh/article/view/183 , DOI: https://doi.org/10.46121/pspc.50.2.4
3. 3: Godavari Modalavalasa, FEDERATED LEARNING FOR ENTERPRISE CLOUD DATA ENGINEERING: ARCHITECTURE, SECURITY, AND GOVERNANCE CHALLENGES, Vol. 51 No. 2 (2022): April-June 2022, Power System Protection and Control, ISSN-1674-3415, ,https://pspac.info/index.php/dlbh/article/view/184, DOI: https://doi.org/10.46121/pspc.51.2.5
4. AI-Driven Document Processing for Customs and Logistics: Automating Millions of Email-Based TransactionsPraveen Kumar Dora Mallareddi, Feroskhan Hasenkhan, Debabrata Das7/26/2022 Newark Journal of Human-Centric AI and Robotics Interaction 3, https://www.njhcair.org/index.php/publication/article/view/13
5. Naresh Lokiny. (2022). Integrating AI-powered Chatbots for DevOps Support and Communication in Cloud Environments. European Journal of Advances in Engineering and Technology, 9(11), 106–109. https://doi.org/10.5281/zenodo.13325989
6. Naresh Lokiny, (2021), "Disaster Recovery and Business Continuity Planning in DevOps Cloud with AI", International Journal of Science and Research (IJSR), 10(3), 2022-2027. https://dx.doi.org/10.21275/SR24724151733, https://www.ijsr.net/getabstract.php?paperid=SR24724151733

DOI: 10.46121/pspc.51.4.10

7. Naresh Lokiny, & Ranganath Nandanampati. (2020). DevSecOps: Integrating Security into DevOps with AI in Cloud. Journal of Scientific and Engineering Research, 7(10), 239–242. https://doi.org/10.5281/zenodo.13348695
8. Naresh Lokiny, & Pradip Reddy. (2021). Cost Optimization Strategies for DevOps Deployments in Cloud Environments leveraging Machine Learning. European Journal of Advances in Engineering and Technology, 8(3), 69–72. https://doi.org/10.5281/zenodo.13325845
9. Jobayar Alom, Ahsan Ahmed. (2022). Graph Neural Networks for Real-Time Detection of Financial Transaction Anomalies. Acta Scientiae, 24(5), 82–90. https://doi.org/10.22178/acta.24.5.6
10. **Kaleshwar Aryasomayajula**, PREVENTING BIAS IN AI-BASED DECISION-MAKING: ANALYZING TECHNIQUES TO REMOVE UNFAIR PREJUDICE IN ALGORITHMIC OUTCOMES, Vol. 50 No. 2 (2022): April-June 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/294
11. Kaleshwar Aryasomayajula, MODERN DEVELOPMENT PRACTICES: INVESTIGATIONS INTO SERVERLESS COMPUTING, LOW-CODE/NO-CODE PLATFORMS, AND DEVELOPER PRODUCTIVITY IN REMOTE ENVIRONMENTS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/364
12. **Vikas Suresh Bagora**, KERNEL-LEVEL PERFORMANCE OPTIMIZATION FOR CARRIER-GRADE BROADBAND AND HIGH-SPEED NETWORKING SYSTEMS, Vol. 47 No. 1 (2019), Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/359
13. **Vikas Suresh Bagora,** SCALABLE 128G FIBRE CHANNEL AND ETHERNET CONVERGENCE FOR NEXT-GENERATION DATA CENTER NETWORKS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/362
14. **Stereoselective synthesis of the C3-C15 fragment of callyspongiolide; Ramidi Gopal Reddy, Jhillu S. Yadav and Debendra K. Mohapatra. (Tetrahedron Letters.** 2018, 59, 3579-3582). https://doi.org/10.1016/j.tetlet.2018.08.041, https://www.sciencedirect.com/science/article/pii/S0040403918310426
15. **Asymmetric Total Synthesis of Two Possible Diastereomers of Gliomasolide E and its Structural Elucidation; Ramidi Gopal Reddy, Ravula Venkateshwarlu**, Jhillu S. Yadav and Debendra K. Mohapatra. (J. Org. Chem. 2017, 82(2), 1053-1063). https://doi.org/10.1021/acs.joc.6b02611 https://pubs.acs.org/doi/abs/10.1021/acs.joc.6b02611
16. **Amanullakhan Pathan Jayadip Ghanshyambhai Tejani, Rahul Shah, Hiral Vaghela, Shailesh, Vajapara , Controlled Synthesis and Characterization of Lanthanum Nanorods,** 2020/5/1, International Journal of Thin Films Science and Technology, Natural Sciences Publishing Corporation (NSP)
17. https://scholar.google.com/citations?view_op=view_citation&hl=en&user=efbNMkwAAAAJ&citation_for_view=efbNMkwAAAAJ:M3NEmzRMIkIC
18. **Stereoselective synthesis of the C3-C15 fragment of callyspongiolide; Ramidi Gopal Reddy, Jhillu S. Yadav and Debendra K. Mohapatra. (Tetrahedron Letters**. 2018, 59, 3579-3582). https://doi.org/10.1016/j.tetlet.2018.08.041, https://www.sciencedirect.com/science/article/pii/S0040403918310426
19. **Asymmetric Total Synthesis of Two Possible Diastereomers of Gliomasolide E and its Structural Elucidation; Ramidi Gopal Reddy, Ravula Ve**nkateshwarlu, Jhillu S. Yadav and Debendra K. Mohapatra. (J. Org. Chem. 2017, 82(2), 1053-1063). https://doi.org/10.1021/acs.joc.6b02611, , https://pubs.acs.org/doi/abs/10.1021/acs.joc.6b02611
20. **Kaleshwar Aryasomayajula**, PREVENTING BIAS IN AI-BASED DECISION-MAKING: ANALYZING TECHNIQUES TO REMOVE UNFAIR PREJUDICE IN ALGORITHMIC OUTCOMES, Vol. 50 No. 2 (2022): April-June 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/294
21. **Kaleshwar Aryasomayajula, MODERN DEVELOPMENT PRACTICES: INVESTIGATIONS INTO SERVERLESS** COMPUTING, LOW-CODE/NO-CODE PLATFORMS, AND DEVELOPER PRODUCTIVITY IN REMOTE ENVIRONMENTS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/364

DOI: 10.46121/pspc.51.4.10

22. **Controlled Synthesis and Characterization of Lanthanum Nanorods**, **Author(s), Jayadeep Tejani, Rahul Shah, Hiral Vaghela, Shailesh Vajapara, Amanullakhan Pathan**, doi:10.18576/ijtfst/090205, URL: https://www.naturalspublishing.com/Article.asp?ArtcID=20705, **May-2020**
23. Vetter, J. S., Brightwell, R., Gokhale, M. et al. (2022) 'Extreme heterogeneity 2022: Fostering post-exascale computing', *International Journal of High Performance Computing Applications*, 37(6), pp. 555–578.
24. Atchley, S., Zimmer, C., Lange, J. et al. (2022) 'Frontier: Exploring exascale', *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pp. 1–16.
25. Bates, N., Ghatikar, G., Abdulla, G. et al. (2022) 'Electrical grid and supercomputing centers: An investigative analysis of emerging opportunities and challenges', *Informatik Spektrum*, 45(1), pp. 34–47.
26. Sîrbu, A. and Babaoglu, O. (2022) 'Predictive analytics for HPC workload and power management: A production case study', *IEEE Transactions on Cloud Computing*, 12(1), pp. 234–248.
27. Klusáček, D., Soysal, M. and Suter, F. (2022) 'Alea and beyond: A decade of HPC workload simulation', *Journal of Parallel and Distributed Computing*, 162, pp. 34–52.
28. Patki, T., Thiagarajan, J. J., Ayala, A. and Islam, T. (2022) 'Performance optimality or reproducibility: That is the question for exascale power management', *IEEE Transactions on Parallel and Distributed Systems*, 34(5), pp. 1421–1436.
29. Reed, D. A., Gannon, D. and Dongarra, J. (2022) 'Reinventing high-performance computing: Challenges and opportunities in the AI era', *Communications of the ACM*, 66(2), pp. 34–43.
30. Etinski, M., Corbalán, J., Labarta, J. and Valero, M. (2020) 'Understanding the future of energy-performance trade-off via DVFS in HPC environments', *Journal of Parallel and Distributed Computing*, 140, pp. 91–104.
31. Sakamoto, R., Cao, T., Kondo, M. et al. (2022) 'Production hardware overprovisioning: Real-world power constraints and their implications', *IEEE Transactions on Parallel and Distributed Systems*, 33(9), pp. 2148–2162.
32. Eastep, J., Sylvester, S., Cantalupo, C. et al. (2021) 'Global extensible open power manager: A vehicle for HPC community collaboration on co-designed energy management solutions', *International Journal of High Performance Computing Applications*, 35(3), pp. 247–266.
33. Krzywaniak, A., Czarnul, P. and Proficz, J. (2022) 'GPU power capping for energy-performance trade-offs in training of deep convolutional neural networks for image recognition', *Journal of Computational Science*, 67, pp. 101–116.
34. Chen, H., Ma, X., Hariri, S. and Youssif, A. (2022) 'Energy-aware backfilling strategies for job scheduling in high-performance computing', *IEEE Transactions on Sustainable Computing*, 8(2), pp. 289–302.
35. Patel, T., Wagenhäuser, A., Eibel, C. et al. (2022) 'What does power consumption behavior of HPC jobs reveal? Demystifying, quantifying, and predicting power consumption characteristics', *IEEE Transactions on Sustainable Computing*, 9(1), pp. 78–92.
36. Wyatt, M. R., Herbein, S., Gamblin, T. et al. (2021) 'PRIONN: Predicting runtime and IO using neural networks for HPC job power estimation', *Proceedings of the International Conference on Parallel Processing (ICPP)*, pp. 1–12.
37. Cao, T., Huang, W., He, Y. and Kondo, M. (2022) 'Cooling-aware and power-capped workload scheduling for green HPC data centers', *Future Generation Computer Systems*, 129, pp. 187–201.
38. Rountree, B., Ahn, D. H., de Supinski, B. R. et al. (2020) 'Beyond DVFS: A first look at performance under a hardware-enforced power bound', *IEEE Transactions on Parallel and Distributed Systems*, 31(8), pp. 1806–1819.
39. Bhalachandra, S., Porterfield, A., Olivier, S. L. and Prins, J. F. (2021) 'An adaptive core-specific runtime for energy efficiency', *Concurrency and Computation: Practice and Experience*, 33(19), e6252.

DOI: 10.46121/pspc.51.4.10